HOMEWORK #3:

A Fistfull of Pointers.

Due Date: Friday, September the 27th, 11:59:59 PM
For this assignment, you need to submit a file called '`linkedlist.hpp`'.
Remember to put your name and section at the top of your program file.

Problem

You were so successful with the arraylist software module, that Bender has now been licensing it to spaceships all around the galaxy.
However, some prospective clients have asked Bender for a List implementation that does not uses arrays, as their spaceships are tight on memory. And Bender, in a display of charming salesmanship, has actually promised to deliver such an implementation!
Once again you are called to the rescue. Bender has already squandered the advance payment, and there is no way to know how violent the clients will become if the product is not delivered.

Implement a Linked-List version of the List ADT, and show Bender that you can both save memory and wrangle a whole herd of wild pointers. A '`linkedlist.h`' has been provided for you with the specification of the promised data structure.

Testing

Use the provided tester files to check if your implementation is working correctly.

- The program ‘large`tester.cpp`’ uses the LinkedList class and the intended output is ‘`largeoutput.txt`’.

Useful Hints

1. Carefully read the comments of each member function.
2. Write down an algorithm for the function before you start coding it.
3. Develop your member functions one at a time, starting from the simplest ones.
   Move to the next function only after the previous one has been tested.
   Trying to code the whole class and then remove the bugs may prove to be too big a task.
4. Suggestion: Implement ‘`insert_front()`’ first.
5. Use the provided friend function `operator<<` to observe the status of your lists.
6. When a function that needs to return something “panics”, return a value as specified in the comments.
7. There is an **auxiliary constructor available**, which is handy for the implementation of the insertion function.